Pixel-wise Semantic Segmentation Algorithm Based on Deconvolution Network

Hui Li
School of Information Science and Technology, Huizhou University, Huizhou 516000, Guangdong, China

Xiaowei Hu, Xuemiao Xu*, Xuehan Yuan
School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China
*Corresponding author (E-mail: xuemx@scut.edu.cn)

Abstract
The performance of semantic segmentation will directly affect the precision of the following image analysis, recognition and etc. To deal with the issues existed in semantic segmentation, such as the edge discontinuities, inaccurate region segmentation, this paper proposed a pixel-wise semantic segmentation algorithm based on deconvolution network. The proposed approach detects the edge with a structured method and generates a pixel-wise probability map with the spatial information, which indicates the probability of each pixel that belongs to one of the predefined classes. Based on this probability map, edge search and pixel-wise region partition is applied to get a fine segmentation. Experiments showed that the proposed method outperforms the semantic segmentation approach based on the existing deconvolution network with higher accuracy and more edge details. And meantime the algorithm has good robustness and real-time performance.

Key words: Deconvolution Network, Pixel-wise, Semantic Segmentation, Contour Extraction.

1. INTRODUCTION

Semantic segmentation is the task of partitioning an image to different parts together which belong to the same object class. Semantic information is extracted from the captured image and used to understand the content of the image and help the following tasks such as recognition, classification and analysis. In recent years, convolutional neural network is a hot topic of computer vision and has gained great improvements in vision tasks such as image classification (Simonyan and Zisserman, 2014; Szegedy and Liu, 2015), object detection (Girshick and Donahue, 2014; He and Zhang, 2014; Girshick, 2015; Ren and He, 2015), tracking (Hong and You, 2015; Li and Porikli, 2014), semantic segmentation (Chen and Papandreou, 2014; Dai and He, 2015; Dai and He, 2015; Long and Shelhamer, 2015).

Recently, Noh H et al. (Noh and Hong, 2015) proposed a semantic segmentation algorithm based on deconvolution network, this algorithm works well in objects in multiple scales and images with detail structures. Though they have achieved the best performance on PASCAL VOC 2012 dataset, their approach still can only give a rough segmentation results and cannot provide the exact edge map. As shown in Figure 1, the feet of the cat and the branch cannot be well captured, and the reconstructed image has lost some information existed in the input image.

![Figure 1. The Results of the Deconvolution Network](image-url)

To solve the aforementioned issues of edge over fitting, inaccurate region segmentation, we proposed a pixel-wise semantic segmentation algorithm based on deconvolution network. The proposed approach can be seen as a optimization of the algorithm proposed by Noh H (Noh and Hong, 2015) with a refinement of the
probability map and a structured method to detect the edge. Our main contribution can be summarized as follows:

1) Based on the original semantic segmentation algorithm on deconvolutional network and a rough edge detection of the input image, a more accuracy probability map can be generated. The rough detected edge can help adjust the output edge contour of the deconvolution network and thus a fine segmentation probability map is obtained.

2) The probability map indicates the probability of each pixel that belongs to one of the predefined classes. From the refined probability map, several closed segmentation contour can be generated. With a further semantic filtering, the final semantic segmentation can be obtained.

The rest of the paper is organized as follows: section 2 introduces the related work; Section 3 describes the pixel-wise semantic segmentation algorithm based on deconvolution network; Section 4 gives the experimental results and some analysis, Section 5 concludes this paper.

2. RELATED WORK

Having a semantic segmentation is a big advantage when trying to analyze and understand the image. Traditional image segmentation includes the threshold method, region segmentation, edge detection and etc. (Bali and Singh, 2015). Recently a new approach has been proposed, Rahmanni et.al. (Rahmani and Akbarizadeh, 2015) proposed a unsupervised feature learning method based on spectral clustering, the proposed method can avoid the instability of the clustering but lead to wrong labeling. Kalinin et.al. (Kalinin and Sirota, 2015) proposed to use the graph cut to partition the super-pixel, and this method generates too much regions and hard to aggregation. Kim et.al (Kim and Yoo, 2015) proposed a higher order correlation method to merge the super-pixel, though they can give a good segmentation results, however, their approach is very time- consuming. Zhou et.al. (Zhou and Liu, 2014) proposed to use the salience detection to give a union segmentation, and this method works bad in complex background and multi-object images.

Semantic segmentation combined the traditional image segmentation and object detection together. Recently most of the related work is based on convolution neural network due to its good performance. Dai J et.al. (Dai and He, 2015) proposed to use a bounding box to guild the network do the semantic segmentation, and this approach can effetely deal with the labeling and training process with the bounding box. Long J et.al. (Long and Shelhamer, 2015) build an end to end, pixels to pixels network that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning.

Noh H et.al. (Noh and Hong, 2015) proposed an instance-wise semantic segmentation by learning a deconvolution network. The network takes a sub-image potentially containing objects as an input and produces pixel-wise class prediction as an output. The final semantic segmentation of the whole image is obtained by extracting each candidate proposal from the image and aggregating the proposals to the original image space.

3. PIXEL-WISE SEMANTIC SEGMENTATION BASED ON DECONVOLUTION NETWORK

Convolutional neural network can extract the high-level features of the object, the features extracted are very representative and produce a superior segmentation to those non-deep-learning methods. However, the convolutional or deconvolution network always contains the pooling and unpooling layers to deal the feature map, so the missing of the edge information of the object is unavoidable. The final segmentation can only give the approximate position of the target without details. In this paper we will optimize the existed algorithm and get a more precision segmentation. To do this we proposed a pixel-wise semantic segmentation based on deconvolution network. The main algorithm will give us a precision probability map and accurate segmented regions.

3.1. Probability Map for Pixel-Wise Semantic Segmentation

In the deconvolution network, the pooling and unpooling are used repeatedly, which will lead to the weaken of the edge information. To recover the missing information, more prior should be added. Luckily the edge detection has become a mature field and lots of successful edge detection has been proposed. The missing information can be recovered by leveraging the edge detected by some existed edge detectors. To refine the probability map generated by the deconvolution network, the edge detected by the detectors can used as a guild map, the pixels which are close to the edge map will have a higher probability and lower otherwise. Here we used the structured forests (Dollár and Zitnick, 2013; 2015) to give the exact image contour, and the probability map generated by the deconvolution network provided a global edge probability. Based on the fine and coarse edge information, a fine probability can be obtained. Figure 2 showed the generating process of the pixel-wise probability map.
As shown in Figure 2(b), rough image segmentation can be obtained by deconvolution network, and image contours with binaryzation can be extracted from the segmentation. Define \( D^c_{xy} \) as the pixel value in the position \((x,y)\) of class \( C \), \( D^c_{xy} \in \{0,255\} \), when \( D^c_{xy} = 255 \), denote it as \( D^c_{xy}^+ \), indicating that the position \((x,y)\) has a large probability as the exact image edge, when \( D^c_{xy} = 0 \), denote it as \( D^c_{xy}^- \). Suppose \((x',y')\) is the pixel in the contour, when \( D^c_{xy}^- = 0 \) and \( D^c_{x'y'} = 255 \), and the position of \((x,y)\) and \((x',y')\) are close to each other, then \((x,y)\) will also has a large probability of being the edge. And vice versa, if the pixel \((x,y)\) are far from the contour generated by the deconvolution network, then it has a low probability of being the edge.

To reflect the relationship between the edge and the probability map, here we used the sigmoid function with S shape as the rectified function, which is defined as follows:

\[
S(x,y) = \left( -\frac{1}{1+e^{-\alpha r(x,y)+\beta}} + 1 \right) \times \frac{1+e\beta}{e\beta} \tag{1}
\]

where \( r(x,y) \) is the Euclidean distance between the pixel of position \((x,y)\) and the nearest pixel \((x',y')\) in the contour, which can be denoted as \( D^c_{xy} \), and \( S(x,y) \) is the refined probability of the true edge. The parameter \( \alpha \) is the steepness of the curve, the larger \( \alpha \) is, the steeper the curve is. And when \( \alpha \) becomes large enough, the function can be seen as a linear function, indicating that the probability was decreasing with the increasing distance. In the extreme case, when \( \alpha \to +\infty \), then \( S(x,y) \to 0 \). The parameter \( \beta \) indicates the translation along the x-axis, the larger \( \beta \) is, the the S curve will have a larger translation to the right side, which means that the effect of the contour generated by the deconvolution network will be more on the final true edge map. The most right part of the function, \( \frac{1+e\beta}{e\beta} \), is the normalization term and projects the range of \( S(x,y) \) to \([0,1]\).

At the meantime, to get the true edge information of the original image, the structured forest method proposed by Dollár P et al. ((Dollár P and Zitnick C, 2013; 2015)) is adopted to detect the edge and generate the probability edge map. As shown in Figure 2, each pixel has been labeled the probability of being the edge. With
this probability map, we can combine it with the probability map generated by the deconvolution network, and obtain the final probability edge map with the formulation as follows:

\[
Q(x,y) = H(x,y) \times S(x,y)
\]  

(2)

where \(H(x,y)\) is the probability map generated by structured forest, and \(Q(x,y)\) is final probability map. Here the multiplication operation means that when there is one zero in other term then even it is very near to the contour generated by the deconvolution network, it cannot be the true edge. With these two probability map combined together, a more precision probability can be obtained, as shown in Figure 2(e).

3.2. The Access for Pixel-Wise Segmented Region

Though we have a refined probability map, it is still not enough to get the precision segmented region. To get a more accurate result, we proposed a segment area based method to obtain the contour. The proposed method includes two steps: the first is to select several closed segment contour with the probability map, the second is to do the semantic filtering on the results generated by the deconvolution network, and then we can get the final semantic segmentation.

To get the closed segmented contour, first, the magnitude and orientation of the gradient of the probability map are computed with Sobel operator. The mask of the Sobel operator is as follows:

\[
G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}
\]  

(3)

where \(G_x\) and \(G_y\) are two masks which are used to compute the horizontal and the vertical derivative approximations with convolution operator respectively. And the magnitude of the gradient is computed as follows:

\[
G = \sqrt{G_x^2 + G_y^2}
\]  

(4)

And the gradient’s direction is calculated as follows:

\[
\theta = \text{atan2}(G_x, G_y)
\]  

(5)

Now we have the magnitude and the direction map of the gradient, to get a closed contour, the pixel with the largest magnitude was selected as the start point, and walking to the pixel which has the perpendicular gradient direction until to the start point, then we will get a closed contour.

However, the aforementioned way of finding the closed contour doesn’t guarantee the acquisition of the closed contour. To enhance the robustness of the searching, avoid walking to the breakpoint and to obtain several closed contour from one single probability map, the pixel with submaximal magnitude is used as a second choice. For each patch centered at the current pixel with size 3*3, we will define a principal gradient direction and a secondary gradient direction. First an 8-bin gradient statistic histogram was created. The histogram channels are evenly spread over 0 to 360 degrees with 8 histogram channels. Then we will have a histogram of the gradient direction and the magnitude. The principal gradient direction is defined as the mean value of the magnitudes in the histogram channel with the largest magnitude, which is denoted as \(G_{\text{max}}(x,y)\), indicating the peak value of the gradient direction at pixel \((x,y)\). The secondary gradient direction is defined as the direction with magnitude larger than eighty percent of the peak value if exists.

With the help of the secondary gradient direction, the algorithm of searching the closed contour can be refined. When there is no pixel to continue as the perpendicular direction one, then we turn back the way we came and mark the pixel in the path to be zero, when we come to the pixel with the secondary gradient direction, then restart with this pixel and the next pixel is the one which has the perpendicular gradient direction just as the original algorithm. At the meantime, to obtain more closed contours, even we have a closed contour, we can still travel back from the last pixel in the path to the one with the second gradient direction and keep moving until we get another closed contour and until all the pixel with the second gradient direction has been searched. Then the pixels left on the image which are not on the contours will be recalculated and used to find other contours until the largest probability remained in the image are less than \(\varepsilon\). Finally, we will get many contours, each two contours will be compared and if there are more than \(\gamma\) pixels overlaps, then the non-overlapped one will be grouped to see whether it will be a closed contour. The closed contours generated by the probability map are shown in Figure 3.
As shown in the Figure 3, there are three candidate contour regions. For each contour region, there will be an evaluation and the one with the highest score will be the final semantic segmentation. The score for each contour $C_i$ is computed as follows:

$$E(C_i) = \sum_{j=1}^{l_i} Q(x_j, y_j) / l_i + \alpha \cdot \frac{l_i}{R} + \beta \cdot \sqrt{(\bar{x} - \bar{x}_{D})^2 + (\bar{y} - \bar{y}_{D})^2}$$  \hspace{1cm} (6)$$

where $Q(x, y)$ comes from the probability map of the deconvolution network, indicating the probability of the pixel being the target class, $l_i$ is the number of all the pixels in the contour region $C_i$, $R$ is the number of pixels which is considered to be the target class in the deconvolution network, $(\bar{x}, \bar{y})$ is the center of the contour $C_i$, $(\bar{x}_{D}, \bar{y}_{D})$ is the center of the segmented region in the deconvolution network, which is shown in Figure 2(b). The first term in $E(C_i)$ describes the average score of all the pixels in the contour that belongs to the target class, the second term represents the ratio of the area of the contour to the area of the segmented region in deconvolution network, which can remove small contours, the third terms is about the distance between the center of the contour and the one of the segmented region in deconvolution network, which can be used to remove the contours lying in the border of the image. The parameter $\alpha$, $\beta$ is used as a balance weight. The contour with the highest score will be used as the final pixel-wise semantic segmentation.

4. EXPERIMENTAL RESULTS

The proposed approach is based on the semantic segmentation algorithm with deconvolution network ((Noh and Hong, 2015)), and an elaborate one with pixel-wise segmentation. The experimental results are shown in Figure 4-8. The proposed method is compared to two existing semantic segmentation methods, including the deconvolution network based method proposed by Noh H et al. ((Noh and Hong, 2015)), the saliency detection based method proposed by Zhou C et al. ((Zhou and Liu, 2014)). As shown in Figure 4, the results of Zhou C et al.’s method can roughly segment the target region but there exists the issue of contour conglutinations. The approach proposed by Noh H et al. can effectively detect the foreground but due to the pooling and unpooling operation, the segmented contour may lose some details. However, with our pixel-wise semantic segmentation algorithm, we can obtain image contour with higher accuracy.
To give a better and objective evaluation on the semantic segmentation, the Precision Rate P and the Recall Rate R are used as the quantitative measurement. The original input image is used as the ground truth and each pixel is labeled as one predefined class. P is defined as the fraction of the labeled pixels in the segmentation that are belong to the particular class. R is defined to be the fraction of the pixels that belong to the particular class that are successfully labeled in the segmentation. The F1 score considers both the precision and recall are defined as follows:

$$F_1 \text{ score} = \frac{2PR}{P+R}$$ (7)

In the experiments, 500 images are used as the test set, the average performance of the proposed method, the deconvolution network based method and the saliency detection based method on the precision rate P, the recall rate R and the F1 score are listed in Table 1. As shown in Table 1, the proposed method outperforms other two methods in terms of the recall ratio and the F1 score.

Table 1. Comparison of different semantic segmentation methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>The proposed method</td>
<td>0.9432</td>
<td><strong>0.9899</strong></td>
<td><strong>0.9659</strong></td>
</tr>
<tr>
<td>Noh H et al.</td>
<td>0.9299</td>
<td>0.9881</td>
<td>0.9581</td>
</tr>
<tr>
<td>Zhou C et al.</td>
<td><strong>0.9655</strong></td>
<td>0.9053</td>
<td>0.9344</td>
</tr>
</tbody>
</table>

5. CONCLUSION
The proposed pixel-wise semantic segmentation can effectively overcome the inaccurate segmentation of the existed algorithms with more edge details. The experimental results show that the proposed method provides a better segmentation both in visual quality and quantitative measurement. However, the proposed approach cannot handle images with too many lines and images without obvious foreground, which will be studied in the future work.

Acknowledgements
This work was supported by NSFC (Grant No.61272293), Science and Technology Planning Major Project of Guangdong Province (Grant No.2015A070711001), and the Field Scientific Research Project of Huizhou University (Grant No. hzuxl201615).

REFERENCES


